## Remarks

Summary: By this Response, arguments are presented for the patentability of the pending claims. These arguments are necessitated by the grounds of rejection set forth for the first time in the Final Action

Response to Paragraph 9: Consideration of amended claims 1, 7, 15, and 17-20 as being patentable over Chung in view of Bowman is respectfully requested. The content of these claims was specified in the last Amendment and Response, to which reference is now made.

The Final Action kindly specifies the details of the rejections, paragraph-by-paragraph of each claim. However, thc following respectfully indicates that many of the premises on which thc current rejections are based, are not supported by the disclosures of the cited references. Because the rejection of claim 1 is repeated with respect to every other claim, these remarks reference the claim 1 rejection, and apply to the other claims as well.

Premise 1: Chung's Disclosure Is Used For A (Software) Testing Environment: The cite was to C1, L44-55, which references Chung's review of the third party "Purify" testing software. It is respectfully submitted that the subject matter of the quote from Chung was not hat Chung uses its disclosure for. Thus, C1, L44-55 do not teach that Chung uses its Checkpoint and Restoration System for software testing. Rather, at C1, L44-55 Chung makes the observation that even though others have provided software test products (e.g., the Purify system),

> "no amount of verification...testing during the software debugging process will detect and eliminate all software faults..." (C1, L47+)

Chung goes on to say that:

> "It is therefore desirable to provide mechanisms that allow a user application process to recover from a fault with a minimum amount of lost information." (C1, L57+)

The point is that Chung disparages software testing (e.g., using the Purify system), and says that its Checkpoint and Restoration System should be used with a software program even though that software program has already been tested (e.g., by the

Purify system). It is clear then that Chung does not teach use of its system **for a** (software) "testing environment". Rather, Chung says that its system should be used **despite prior testing of the software**. Moreover, the checkpoint and recovery system of Chung is not said by Chung to be applicable to the previous testing (such as by the Purify software), i.e., Chung does not say "use my system to make the prior software testing programs better". In conclusion, by the C1, L44-55 reference to the exemplary Purify system, Chung teaches away from improving the prior software testing systems, and away from use of those "testing environments". In view of the teaching away, Chung does not motivate one to improve any of the prior software testing systems.

Premise 2: Chung Teaches All Aspects Of Claim 1 (b) Except "Test Specific Operations Of A Test Suites": Claim 1 (b) as identified in the Action recites the following, where the bold identifies elements beyond the "Test Specific Operations Of A Test Suites".

> a **plurality of software test systems** configured in a **distributed processing framework,** each of the test systems being configured to operate under **control of a test application program** to test a particular software program with respect to ability to perform specific operations, the **test application program being configured to execute** one of the test suites that defines the tests to be performed on the particular software program, first and second test execution requests for testing the particular software program **being configured to be executed on different ones of the software test systems distributed among the distributed processing framework;**

As indicated in the last response, Chung is characterized by a problem in use of general software, and not in the testing of software under the control of software test application programs enabled to run on a distributed processing framework. Chung focuses on the persistent state in running the general software (C2, L15-21, C2, L45-47, for example). Chung refers to a user application process (C2, L63) which is executed on one machine. The Chung focus is on persistent state in relation to checkpointing, (C3, L1-3, C3, L28-30, and C12, L49), and there is no checkpoint ability corresponding to each of many separate tests, i.e., in the claimed manner of the individual software tests. Further, the many processing units of Chung are not the claimed specific software test systems for test execution requests. For example, the export noted at C5, L10-12 is not related to a test execution request, and instead

relates to recovery at the remote node when recovery at the local node is not successful. In view of these remarks, it is respectfully submitted that this premise is not supported by the Chung disclosure.

Further, if Chung would motivate one to look for any feature of software testing as an addition to its checkpoint and recovery system it would be to look for more than the test specific operations of test suites to meet the claim text. However, as advanced above, Chung teaches away from improving the prior software test systems, and thus would not motivate one to improve any of the prior software testing systems, and thus would **not** motivate one to look for **any** test specific operations of test suites to meet the claim text.

Premise 3: Bowman Teaches The Missing "Test Specific Operations Of A Test Suites": First, the reference to Bowman (Action page 5) is to C83, L1-5 of Bowman. Respectfully, "Version Control" is not stated there. Having reviewed Bowman, it is noted that reference is made to Version Control starting at C59.

As a preface to the discussion of Version Control and testing in Bowman, it is respectfully submitted that Bowman is in effect a textbook on software development. As such, Bowman has **numerous general statements** as to what should be done to develop software. Bowman also has many detailed statements, and as to OOP and CIA (C3-C4) at C5, L15+ concludes that:

> "...OOP allows the software developer to design and implement a computer program that is a model of some aspects of reality,...the ...developer can create an object which can be used as a component in a larger...project in the future."

Significantly, Applicants have been able to identify no teaching in Bowman that the OPP or CIA software architecture should be used in any of the software testing to which Bowman makes general or more specific reference.

Secondly, considering the asserted teaching of Bowman that fills in the missing Chung disclosure, as asserted with respect to claim 1(b), the C59 text describes "Version Control" as a tool (C59, L52), and continues to C61. At C61, L1, under "Product Considerations", a question is raised: "a) Does The Tool Provide

Capabilities to Cater for a System Running on Multiple Platforms or a Distributed System?" It is here that the Action quote attributed to C83 is stated, and states:

> "Ideally, the Version Control tool must be able to operate on all the platforms in use, whilst at the same time performing version control for all components across the entire system."

It is respectfully submitted that the quoted text does not teach the asserted "Test Specific Operations Of A Test Suites" that was admitted as being missing from Chung. First, "Version Control", and not testing of software, is the subject of the quote. Second, "Version Control" is defined (C59, L52+) as "...tools **control access to source code** as it is developed and tested, and allow multiple versions to be created...". Also, the state of the system (C60, L9) may be recorded. Significantly, in the various descriptions of Version Control, Bowman does **not** say that Version Control is a **way of testing** that is to be done across all platforms. Further, Bowman does **not** there say that testing, itself, is to be done across all platforms. Rather, in describing Version Control's function in controlling access to source code (C59, L52+), none of the details of the identified claim 1(b) clause are specified:

> a **plurality of software test systems**...each of the test systems being configured to operate under **control of a test application program** to test a particular software program...the **test application program being configured to execute** one of the test suites that defines the tests to be performed on the particular software program, first and second test execution requests for testing the particular software program **being configured to be executed on different ones of the software test systems distributed among the distributed processing framework;**

Further, while Bowman describes some aspects of software testing at C39 and C59, it is respectfully submitted that the Bowman software testing descriptions do not teach the asserted "Test Specific Operations Of A Test Suites" that was admitted as being missing from Chung. Despite those software testing descriptions, Version Control is not described as assisting that software testing other than remotely (and not as claimed) by the noted access to source code (C59, L52+), and other than by recording the state of the system (in general and not the state of testing, C60, L8-10). As is advanced above, the access to source code access does not provide the claimed details of software testing. Rather, in proper context, the quote at C61, L4-7, merely

confirms that (via Version Control) the access to source code must be done across the entire software system being developed.

In an endeavor to distinguish over other testing aspects of Bowman, reference is made to C63, L 48+ in re migration control, and control of multiple versions of source code, etc., as they are "changed, tested, and moved from one development environment into another." This general reference to testing, and movement, does not describe or suggest the detailed text of claim 1 (b) set forth above. It is the version of the software under development that is said to be tested, at a development environment, and no way of doing such testing is described, as contrasts to the claimed way.

In the same endeavor, reference is made to the C39-41 descriptions of software testing. **In contrast to** the claimed software test system being configured to test one particular software program, and the claimed first and second test execution requests configured to be executed on different ones of the software test systems distributed among the distributed framework, Bowman teaches C40, L1-5 that:

> "...for each platform that is supported by the system, there must be a separate set of tests."

The Bowman teaching of one set of tests at each platform clearly indicates that the tests of a particular set of tests are **not to migrate**, as claimed, from one platform to another. Bowman here teaches away from the claimed (claim 1) "first and second test execution requests...being configured to be executed on different ones of the software test systems distributed among the distributed framework,". As to the 35 USC Section 103 standard, when a reference to be combined teaches away in this manner, the combination is not proper.

In the same endeavor, reference is made to Bowman's description of monitoring. At C67, L1-7 the monitoring is generic, and not in any detail corresponding to the claimed (claim 1)

> the respective located software test system executing a respective test execution request being configured to monitor progress in the execution of the respective test execution request on the respective located software test system,

for example. The reference in Bowman to "resources" here is not the claimed monitoring of software tests. As to C68, the Version Control and the V-model are references to "development stages" and not to the claimed details of software testing. Similarly, the C68, L50+ references to testing are generic as to the testing of one stage, and not to the claimed testing. Also, the source code debugger at C96 is described with respect to generic testing (the need to test), and not the claimed testing. Similarly, the C100-104 descriptions are generic guides to testing, and not the claimed testing.

It is respectfully submitted that one skilled in the art, looking at these various general aspects of Bowman, would not be taught toward the claimed testing because there is no teaching or suggestion of the details of the claimed restoring during testing. Moreover, the Bowman references to fault management and recovery (C111, L43+, and C112) are also generic and lack the details set forth in the claims. Even the reference to failure control (e.g., at C108, L35+) is tentative and suggests the need for human intervention...(C112, L57). Moreover, the mere fact that the Bowman software development runs in a distributed environment, does not teach the claimed operations defining how to test the developed software in that environment.

In view of these remarks, it is respectfully submitted that this premise is not supported by the Bowman disclosure.

Premise 4: "Bowman's Teaching of Dividing Software Into Individual Components To View A Computer Program As A Collection Of Largely Autonomous Components Makes It Obvious to Combine Bowman With Chung": The rejection combining Bowman and Chung referenced "largely autonomous components" in Bowman (Page 5). This reference was in a statement of the rationale for combining these two references. By this rationale, Chung was to be supplemented by Bowman, by which the software to be tested was to be divided up (per Bowman) so as to view this one software program as a Bowman "collection of largely autonomous components" (Bowman at C4, L4-5).

Breaking this asserted rationale into its parts, the rationale is understood as follows. Because Bowman teaches (C3, L63+) OOP as being a process of software

development, and because an object is visualized (C4, L1) "as a self-sufficient component that does not require other structure... to perform its specific task" (C4, L3), and because OOP "...views a computer program as a collection of largely autonomous components, called objects, each of which is responsible for a specific task..." (C4, L6), it would have been obvious to divide the one Chung software program into objects, and to do such division without any substantial claimed detail from Bowman, and contrary to Bowman's teachings, to thus provide the claimed details of software testing.

The lack of substantial detail in Bowman as to testing is made clear in the remarks above relating to Premise 3. See for example, the discussion of Version Control and software testing (re C39 and C59).

Moreover, the teaching away by Bowman (away from the claimed tests from one test suite being distributed to other platforms) is made clear by the reference to Bowman at C40, L1-5, where it is stated that:

> "...for **each platform** that is supported by the system, there must be a **separate set of tests**."

The Bowman teaching of **one set** of tests at **each** platform clearly indicates that the tests of a particular set of tests are **not to migrate**, as claimed, from one platform to another. Moreover, this C40 teaching is consistent with the C4 statements about OOP and CIA, namely at C4 (i.e., Bowman saying "an object is visualized (C4, L1) as a self-sufficient component that **does not require** other structure... to perform its specific task (C4, L3)" The Bowman test set stays at the one platform (does not require other structure), and thus Bowman teaches away from the claimed (claim 1) "first and second test execution requests...being configured to be executed **on different ones of** the software test systems distributed among the distributed framework,". As to the 35 USC Section 103 standard, when a reference to be combined teaches away in this manner, the combination is not proper.

Further, it is respectfully submitted that the foregoing is the proper context in which to understand the Bowman "collection of largely autonomous components" (Bowman at C4, L4-5). The use of "autonomous" in respect to the asserted testing of

software according to Bowman is clearly consistent with the Bowman statement that "an object is visualized (C4, L1) as a **self-sufficient** component that does **not require other structure... to perform its specific task** (C4, L3)". If (per the rejections) this statement (of self-sufficiency) is applied to rationalize the filling of the gaps of the generally-defined Bowman software testing, then the test object must be a self-sufficient component that does not require other structure to perform its specific task. To be consistent with the cited Bowman "autonomous components" teaching, this test object would **not** be able to cooperate as claimed at other software test systems distributed among the distributed processing environment. The test object would **not** respond to one test from a test suite that has a second test that is sent to another platform.

Further, this autonomous components" teaching is consistent with the above-noted Bowman cite at C40, L1-5, where it is stated that:

> "...for each platform that is supported by the system, there must be a separate set of tests."

Clearly, "autonomous components" have their own tests, and none of those tests can be configured to be executed on different ones of the software test systems distributed among the distributed processing environment.

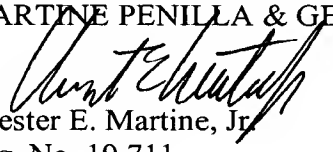
In view of these remarks, it is respectfully submitted that this premise is not supported by the Bowman disclosure.

In view of these remarks, consideration of the pending claims as being patentable over Chung and Bowman is respectfully requested, and allowance of this application is believed to be in order, and is respectfully requested.

Should the Examiner have any questions concerning this Application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,

MARTINE PENILLA & GENCARELLA, LLP



Chester E. Martine, Jr.
Reg. No. 19,711

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
Telephone (408) 774-6908
**Customer No. 32291**